Easy Data Analysis Using R

Marc Paterno

Fermilab

Computing Techniques Seminar
October 25, 2005

What is R?

- "R is a language and environment for statistical computing and graphics."
- R is "not unlike S"
- The author of S, John Chambers, received 1998 ACM Software System Award:

  > *The ACM's citation notes that Dr. Chambers' work "will forever alter the way people analyze, visualize, and manipulate data . . . "*

  Other software systems given this award include:

  1983 Unix, Ken Thompson and Dennis Ritchie
  1991 TCP/IP, Vinton Cerf and Robert Kahn
  1995 World-Wide Web, Tim Berners-Lee and Robert Cailliau
- R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License

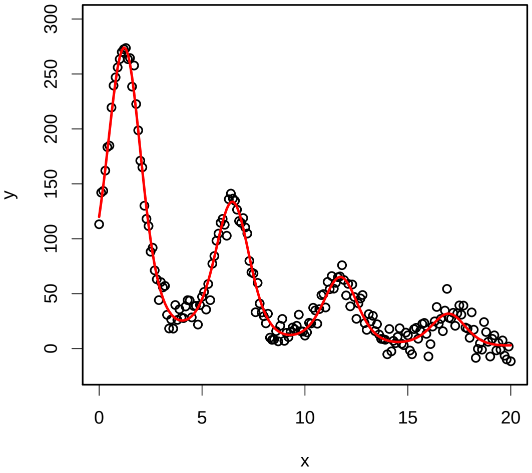
Why use R?

- R provides excellent graphical tools
- The R language is convenient and powerful for data manipulation
- Many modern data analysis techniques are available as R packages

R allows you to concentrate on your data, not on your tools.

R *provides commonly used plot styles*

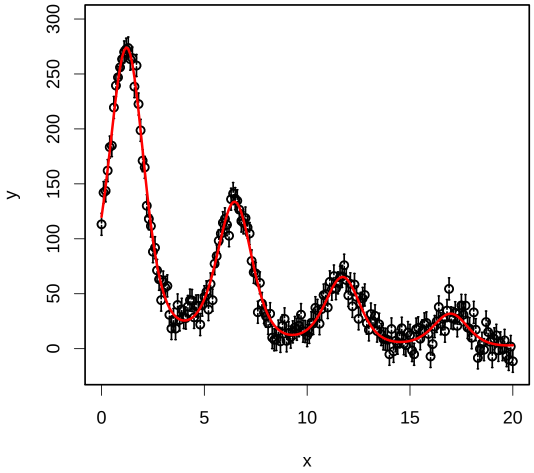
R provides plots we commonly use: *e.g.*, histograms and (x, y) plots



- plots can include fits to data
- many fitting methods supported; shown is the result of (nonlinear) least squares

R *provides commonly used plot styles*

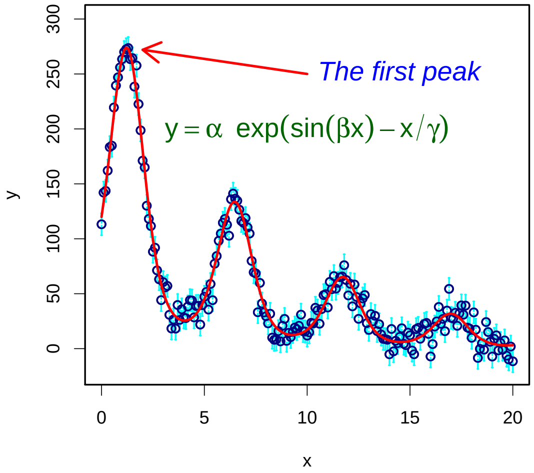
R provides plots we commonly use: *e.g.*, histograms and (x, y) plots



- plots can include fits to data
- many fitting methods supported; shown is the result of (nonlinear) least squares
- . . . and of course plots can include error bars

R *provides commonly used plot styles*

R provides plots we commonly use: *e.g.*, histograms and (x, y) plots



- plots can include fits to data
- many fitting methods supported; shown is the result of (nonlinear) least squares
- . . . and of course plots can include error bars
- . . . and annotations of text, math, symbols
- . . . and color (even to the degree of questionable taste)

R *also provides many useful "modern" plots*

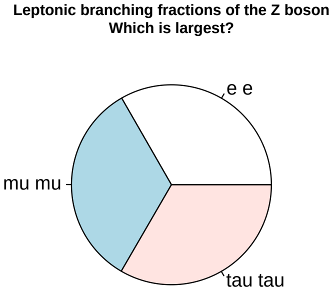
R also provides a variety of useful plot types which are *not* widely known to the physics community, including:

- dot plot: replacement for pie charts and bar charts, studies show dot plots lead to easier perception of patterns
- splom: scatter plot matrix, showing all pairwise correlations for a set of variables
- box-and-whisker plots: for summary comparison of a large number of 1-d distributions
- quantile and QQ plots: for sensitive comparison of two distributions

There are many more special-purpose plots: many statistical tools come with dedicated plot styles (*e.g.,* dendrograms for clustering results).

Good plots help human perception of relationships

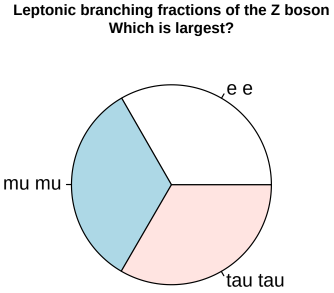
Published studies indicate the human perception is poor at interpreting the pie chart

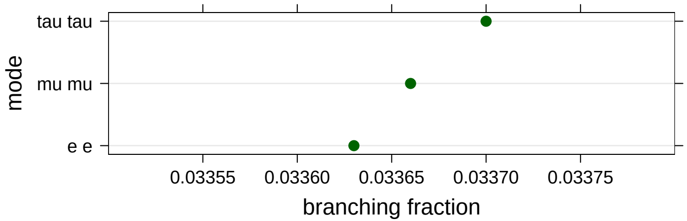


Good plots help human perception of relationships

Published studies indicate the human perception is poor at interpreting the pie chart

The dot plot allows much clearer presentation ...



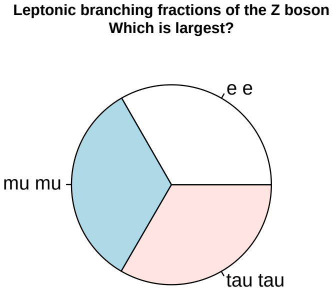


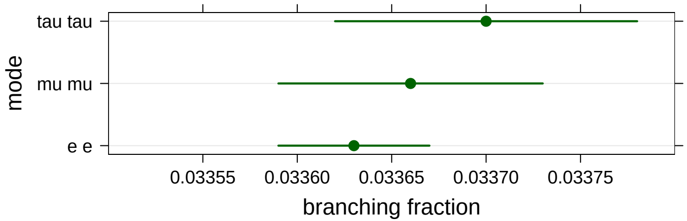
Good plots help human perception of relationships

Published studies indicate the human perception is poor at interpreting the pie chart

The dot plot allows much clearer presentation . . .

And also allows showing error bars

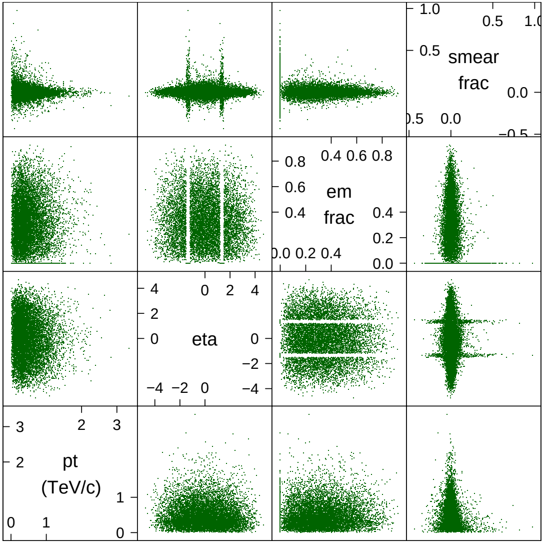




Scatter plot matrix display of toy jet resolution simulation

The scatter plot matrix is a useful device for quickly identifying pairs of quantities with interesting relationships:

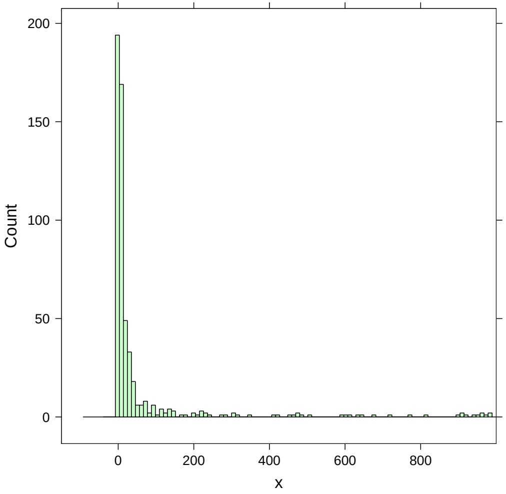
- Shows all pairwise associations between quantities
- Interesting correlations are easily visible
- Unbinned—no features lost due to unfortunate binning



Scatter Plot Matrix

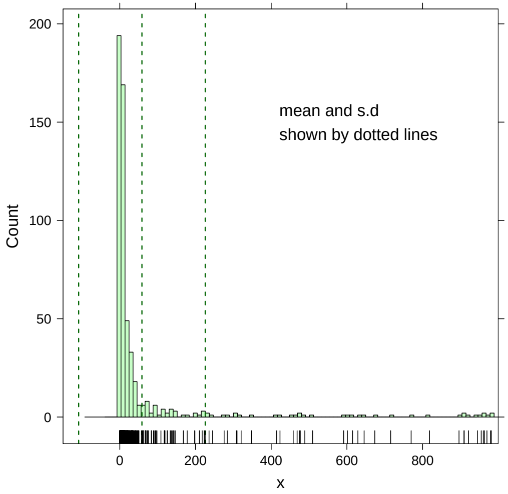
Dealing with asymmetric distributions

- Sometimes we have to deal with data that show “tails”



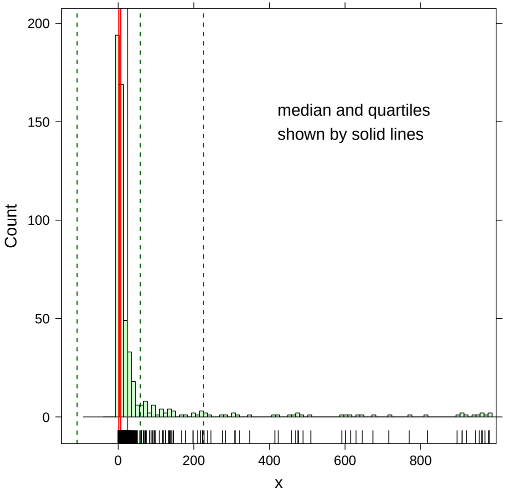
Dealing with asymmetric distributions

- Sometimes we have to deal with data that show “tails”
- For asymmetric distributions, the mean and s.d. can be misleading



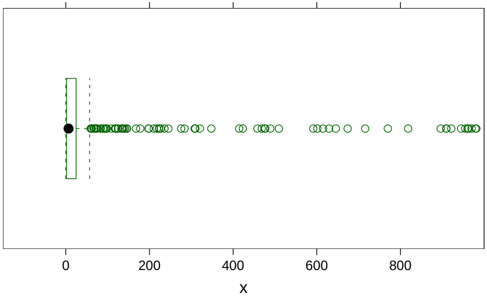
Dealing with asymmetric distributions

- Sometimes we have to deal with data that show “tails”
- For asymmetric distributions, the mean and s.d. can be misleading
- Unless symmetry is known, more informative statistics may be better (*e.g.* median, quartiles)



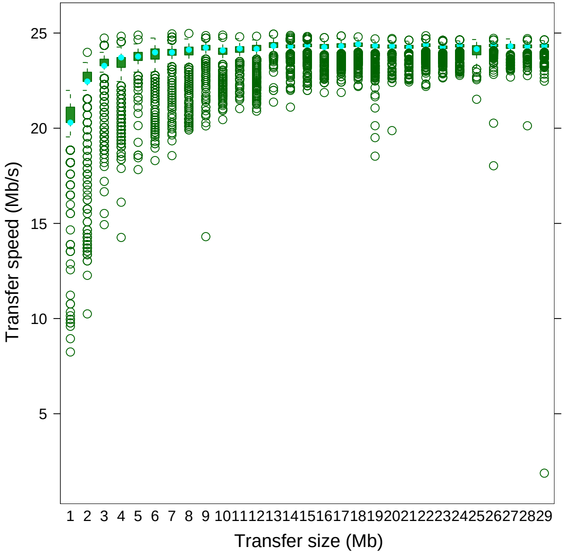
Dealing with asymmetric distributions

- Sometimes we have to deal with data that show “tails”
- For asymmetric distributions, the mean and s.d. can be misleading
- Unless symmetry is known, more informative statistics may be better (*e.g.* median, quartiles)
- Tukey’s box plot summarizes such statistics (and more)



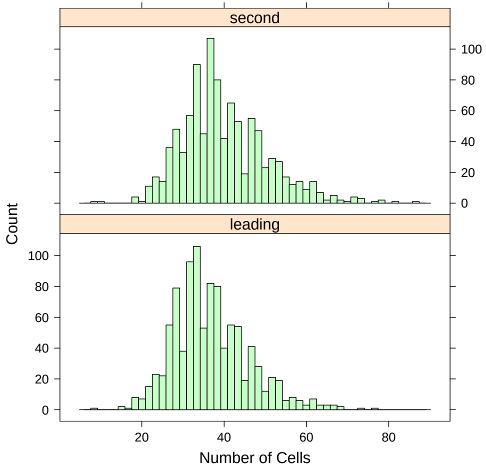
Multiple box plots to show variation across distributions

- We often use the "profile histogram" to summarize the variation in the distribution of *y* as a function of *x*.
- When the distributions are asymmetric, or have outliers, the box plot can be much more informative



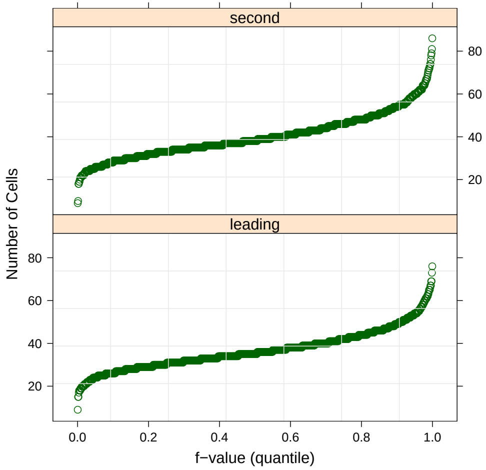
Comparing two (simulated) jet cell multiplicities

- Studies shows human perception is poor at evaluating similar histograms



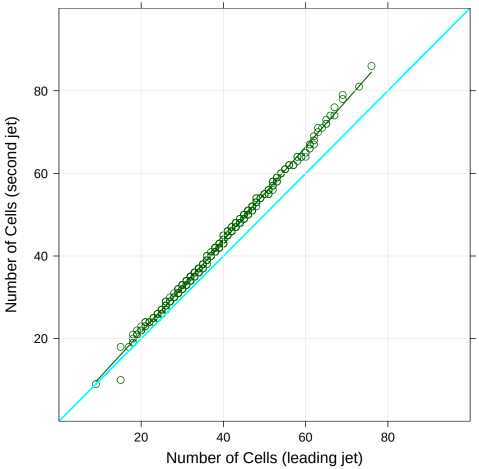
Comparing two (simulated) jet cell multiplicities

- Studies shows human perception is poor at evaluating similar histograms
- Quantile plots (cumulative distributions) are somewhat easier to distinguish



Comparing two (simulated) jet cell multiplicities

- Studies shows human perception is poor at evaluating similar histograms
- Quantile plots (cumulative distributions) are somewhat easier to distinguish
- An quantile-quantile (QQ) plots are easier still
- We clearly see even a small difference: the 2nd jet's NofC distribution has a larger high-end tail



Working with R

- An R session can be saved to disk, and the application state recovered at a later time
- The saved R session is platform neutral: save it on your Linux workstation, move it to your Windows or Mac laptop, and continue work.
- R can read many data formats:
  - text files (local, or *via* URL)
  - common spreadsheet formats
  - Oracle, MySQL, SQLite, PostgreSQL databases (or any ODBC database)
  - DCOM and CORBA
  - many other statistical software systems

Working with R

- An R session can be saved to disk, and the application state recovered at a later time
- The saved R session is platform neutral: save it on your Linux workstation, move it to your Windows or Mac laptop, and continue work.
- R can read many data formats:
    - text files (local, or *via* URL)
    - common spreadsheet formats
    - Oracle, MySQL, SQLite, PostgreSQL databases (or any ODBC database)
    - DCOM and CORBA
    - many other statistical software systems
    - Root trees—local development at Fermilab[1]; allow reading of trees that are "simple", such as those of the CMS experiment's reconstruction framework.

[1] thanks to Adam Lyon

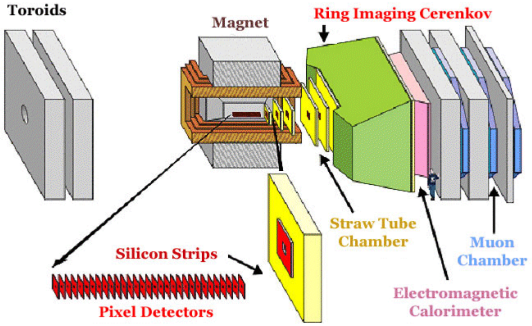
R *package mechanism*

Additional functionality in R comes in through packages

- Distributed package management is integration into the system, similar to:
    - Perl's CPAN
    - the Linux *yum* utility
- Uniform documentation model is observed (helped by package building system)
- Users have all the tools to create (and even distribute) their own packages
- Discovery and installation of new packages is easy
    - Visit `http://cran.r-project.org/` to see what is available
    - Or just use the `install.packages` or `update.packages` functions in R!

Using R *for everyday work (1)*

In working on the DAQ system for the (late) BTeV experiment, we needed to analysis GEANT3 simulation of the pixel detector, to determine how the expected data rate varied with beam luminosity.



Simulation output was converted with a simple Python program to a text file, and read with R:

```
> stations = read.table("btev.dat", ...)
```

This creates a *data frame*, which behaves like a table.

Using R for everyday work (2)

```
> nrow(stations)
[1] 554218

> stations[1:3,]
  nint idx station ntrip
1    1   1       0     0
2    1   1       1     0
3    1   1       2    24
```

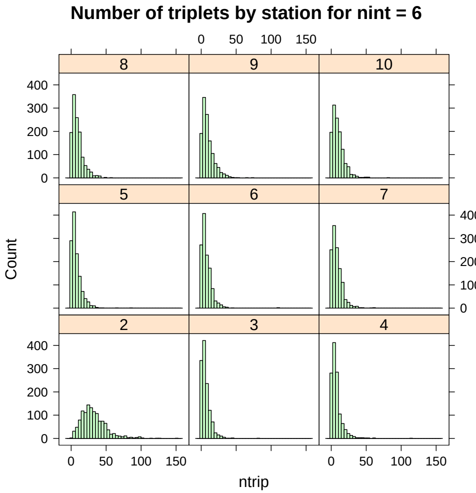
Let’s see the distribution of the number of triplets per station, at a fixed number of interactions:

Using R *for everyday work (2)*

```
> nrow(stations)
[1] 554218

> stations[1:3,]
  nint idx station ntrip
1    1   1       0     0
2    1   1       1     0
3    1   1       2    24
```

Let's see the distribution of the number of triplets per station, at a fixed number of interactions:



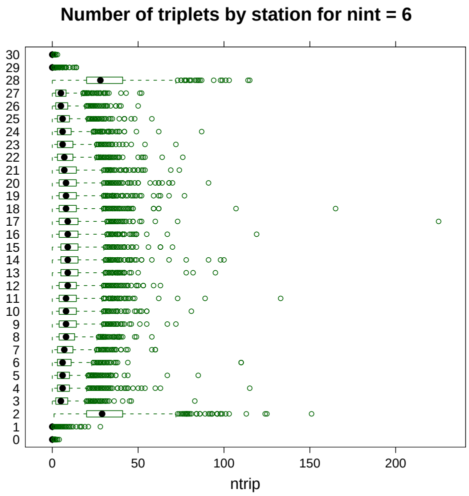
```
> histogram(~ntrip|station, data=stations,
    subset=(station %in% 2:10 & nint==6), ...)
```

Using R for everyday work (2)

```
> nrow(stations)
[1] 554218

> stations[1:3,]
  nint idx station ntrip
1    1   1       0     0
2    1   1       1     0
3    1   1       2    24
```

Let's see the distribution of the number of triplets per station, at a fixed number of interactions:



```
> bwplot(station~ntrip, data=stations,
    subset=(nint==6), horizontal=TRUE, ...)
```

Using R *for everyday work (3)*

Next we want to group data: sum `ntrip` over all stations for each "event", *i.e.* for rows with equal `idx` and `nint`

```
> events=aggregate(stations.ntrip,
           by=list(idx=station$idx,nint=station$nint),
           sum)
```

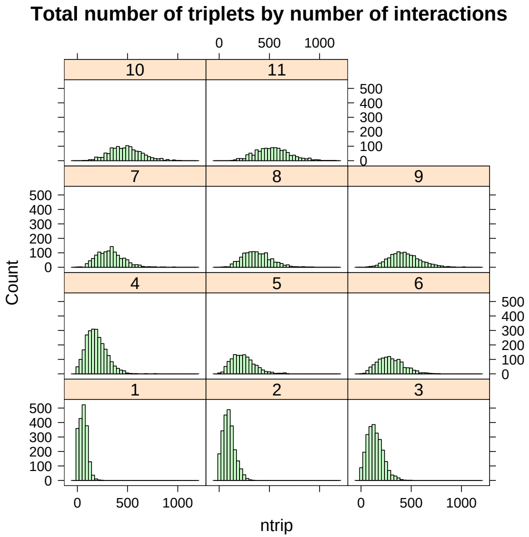
`aggregate` is one of a great many high-level data manipulation tools. And after a little fixing of names, we can print some results:

```
> str(events)
'data.frame':   17878 obs. of  3 variables:
 $ idx: num  1 2 3 4 5 6 7 8 9 11 ...
 $ nint : Ord.factor w/ 11 levels  ...
 $ ntrip: int  226 86 152 8 70 11 3 64 77 117 ...
```

`str` shows the structure of its argument, great for a short summary

Using R *for everyday work (4)*

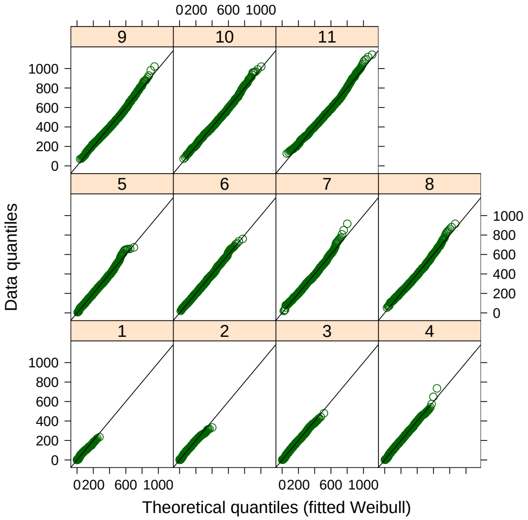
Now we can look at the distributions of total number of triplets in each event



Using R for everyday work (4)

Now we can look at the distributions of total number of triplets in each event

We suspect these may be well-described by the Weibull distribution: let's check, using QQ plots

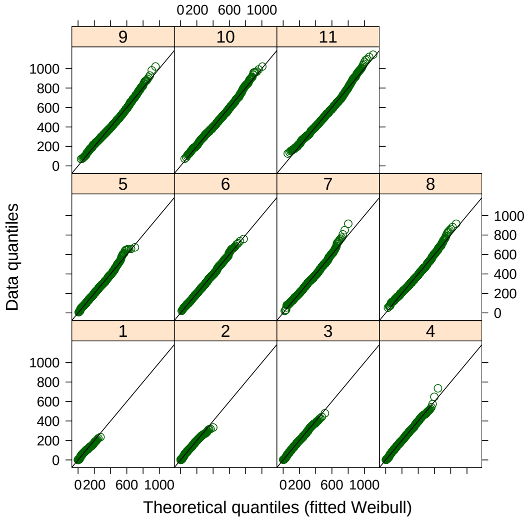


Using R *for everyday work (4)*

Now we can look at the distributions of total number of triplets in each event

We suspect these may be well-described by the Weibull distribution: let's check, using QQ plots

Only at the extreme (1-2%) high tail do the data differ; for purposes of our predictions, we are satisfied the Weibull fits are sufficient

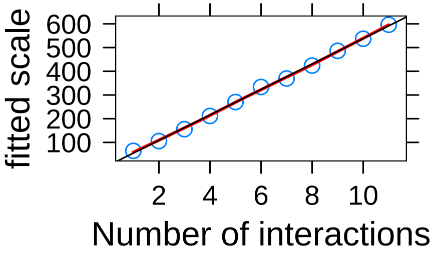


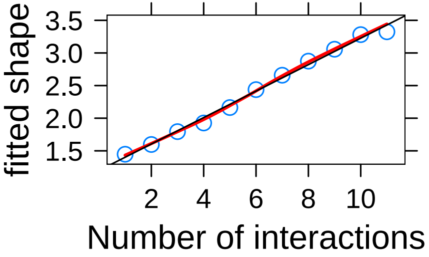
Using R *for everyday work (5)*

Finally, we can capture the fit results, and plot them.

```
    dflist = by(events, events$nint, fit.fromsubframe)
    do.call("rbind", dflist)
```

`by` is another one of the high-level R functions: here it operates on the rows of `events`, grouping them by `nint`, and calls our own function `fit.fromsubframes`, which calls one of R's fitting functions.





Availability of tools

R makes available an enormous variety of statistical tools, *e.g.*

- neural networks
- decision trees
- fitting
- bootstrapping
- clustering
- spatial models
- linear model
- Markov-chain MC
- genetic algorithms

The S language (and so also R) is, more than any other language, the "common tongue" of data analysis.

- Used for reference implementations of many analysis techniques
- As of the time of this writing, there were 590 packages and bundles available in the main repository, CRAN, ...
- ...and there were 122 more at the next largest site (BioConductor)
- Many of these packages present not just one tool, but a large *family* of tools.

Example package: akima

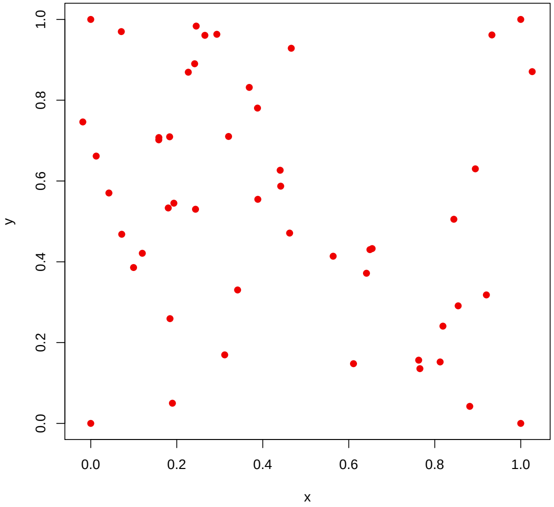
The R package *akima* presents Akima's method[2] for linear or cubic spline interpolation of irregularly spaced data.

[2]Akima, H. (1978), ACM Transactions on Mathematical Software, **4**, 148-164, and Akima, H. (1996), Algorithm 761, ACM Transactions on Mathematical Software, **22**, 362-371.

Example package: akima

The R package *akima* presents Akima's method[2] for linear or cubic spline interpolation of irregularly spaced data.

We begin with irregularly spaced data; we have a z value at each (x, y) point.

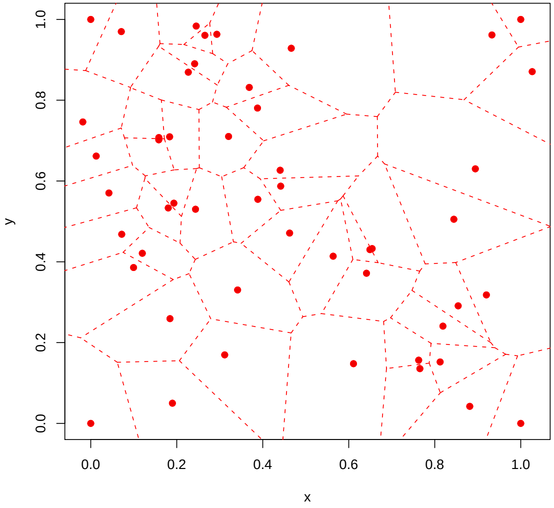


Example package: akima

The R package *akima* presents Akima's method[2] for linear or cubic spline interpolation of irregularly spaced data.

We begin with irregularly spaced data; we have a z value at each (x, y) point.

Akima's method begins with Delaunay triangularization.

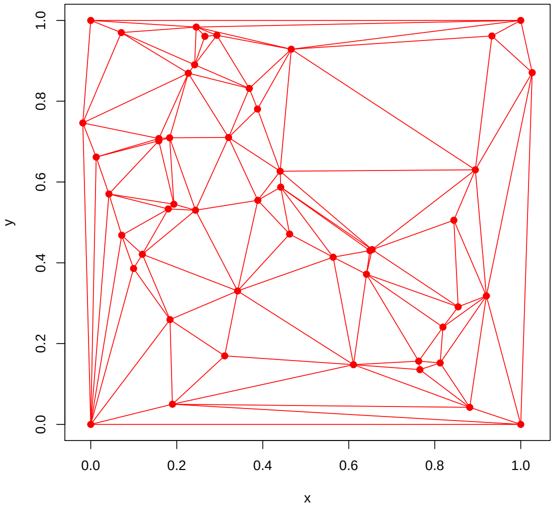


Example package: akima

The R package *akima* presents Akima's method[2] for linear or cubic spline interpolation of irregularly spaced data.

We begin with irregularly spaced data; we have a z value at each (x, y) point.

Akima's method begins with Delaunay triangularization.



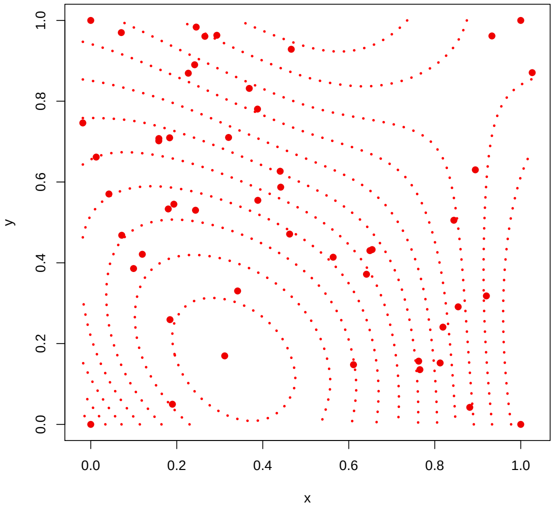
Example package: akima

The R package *akima* presents Akima's method[2] for linear or cubic spline interpolation of irregularly spaced data.

We begin with irregularly spaced data; we have a z value at each (x, y) point.

Akima's method begins with Delaunay triangularization.

Within each triangle, it uses linear or cubic spline interpolation.



Example package: akima

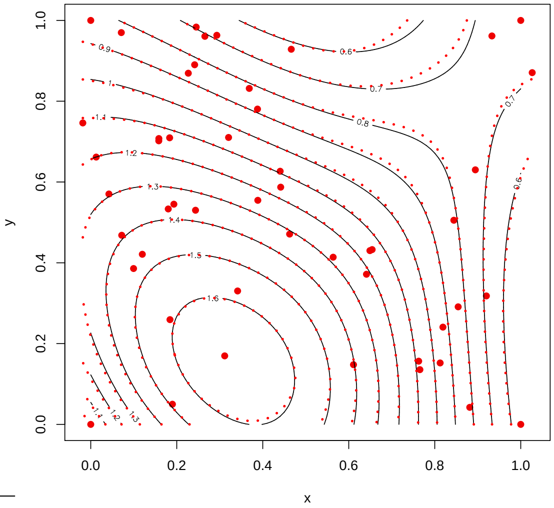
The R package *akima* presents Akima's method[2] for linear or cubic spline interpolation of irregularly spaced data.

We begin with irregularly spaced data; we have a z value at each (x, y) point.

Akima's method begins with Delaunay triangularization.

Within each triangle, it uses linear or cubic spline interpolation.

Even for such sparse data, the accuracy is impressive.



Conclusion

The ease with which you can understand your data is important. My colleagues and I have found R to provide:

1. excellent graphical tools,
2. an easy to learn, powerful, and convenient language for data manipulation, and
3. a host of modern data analysis techniques.

A final point

R allows you to concentrate on your data, not on your tools.

fit.fromsubframe

```r
# Call this with a subselection of the dataframe,
# and it returns a dataframe with the fit to that
# subselection.
fit.fromsubframe <- function(data)
{
  # Fit the Weibull distribution
  data.fit <- fitdistr( data$ntrip, "weibull"
                      , list(shape=1,scale=1))
  wshape <- data.fit$estimate["shape"][[1]]
  wscale <- data.fit$estimate["scale"][[1]]
  data.frame( wshape=wshape, wscale=wscale
            , nint=data$nint[[1]])
}
```